A Summary of CFS I/O Tests.

Zhong C. Lou[1]

Report RNR-90-020, October 1990

NAS Systems Division
NASA Ames Research Center, Mail Stop T045-1
Moffett Field, CA 94035

October 29, 1990

Abstract. We report the results of experiments we performed to test the concurrent files system (CFS) on the Intel Touchstone Gamma Prototype.

[1] The author is with the Mathematics Department, University of California, Berkeley, CA 94720

A Summary of CFS I/O Test

Zhong C. Lou

July 27, 1990

Intel Touchstone Gamma Prototype is a distributed memory MIMD parallel computer based on Intel i860 floating point processor. The system contains 128 compute nodes and 10 I/O nodes. To balance its computation power and the requirement of a high capacity, fast-access mass storage for a large scale parallel application, the system uses a Concurrent File System (CFS) to meet the I/O needs of compute nodes. It is clear the speed of CFS I/O will have an important effect on the overall performance of the machine for a large application.

The I/O subsystem consists of many small disks served by 10 I/O nodes. A group of disks is conneced to each I/O node which has access to the hypercube interconnect of computational nodes. The CFS treats all of the disks in the system as a single logical disk. The system allocates blocks to (from) a file from (to) all of the disks. The result is that the data required by multiple compute nodes, whether from a single file or multiple files, is likely to be on separate disks and can be transfered simultaneously.

In dealing with CFS I/O, at least two bottlenecks must be considered. One is the transfer rate from a compute node to an I/O node which is about 2.4 megabytes/second; the other is the transfer rate from an I/O node to disks which is about 1.0 megabyte/second. If one compute node does the I/O, the first one will limit the maximum transfer rate. If a large number of compute nodes do the I/O, the second one will limit the maximum transfer rate. In the following, we report the results of experiments we performed to test CFS I/O by single and multiple compute nodes under various conditions. All experiments involve writing of a certain number of bytes to one or more CFS files. Since each compute node has a memory of 8 megabytes, the maximum size of write we tried for each node is 6 megabytes.

1. One Compute Node Write to One CFS File

We first test the transfer rate from a single compute node to a CFS file. In figure 1, we display the transfer rate as a function of the total size of writing. The values shown are an average of five times writing. It can be seen that the satuation transfer speed is about 2.2 megabytes/second, which is close to the maximum transfer rate of 2.4 megabytes from a compute node to I/O nodes.

2. Multiple Compute Nodes Write to One CFS File

In figure 2, we display the result of multiple nodes write to one CFS file. In this test, we let each participating compute node write 6 megabytes to a single CFS file. Distinct file pointers are set up for different compute nodes before the writing starts so that different nodes can write to their own places in the file at the same time. We did not allocate a specific number of I/O nodes to be used in this case so we assume that all 10 I/O nodes may be used. The values shown are also an five-time average. The curve showes the changes of transfer rate as a function of compute nodes. Here and thereafter, the transfer rate is defined as

$$\begin{aligned} Transfer\ rate &= \frac{total\ task\ size}{\max T_i} \\ &= \frac{number\ of\ nodes\ \times\ bytes/node\ transfered}{\max T_i} \end{aligned}$$

where T_i is the time taken by the ith compute node to complete its I/O. The maximum is taken over all participating compute nodes. We see that the transfer rate increases as the number of compute nodes increases until it reaches about 9 megabytes/second. This limit of transfer rate is caused by the second bottleneck we mentioned above. We noticed in figure 2 that there is only a small increase between one compute node and two compute nodes. We do not have an explanation for this yet.

3. Multiple Nodes Write to Distinct CFS Files In figure 3, we show results of multiple nodes write to distinct files. Tests were made to let each node write 1, 2, 4, 6 megabytes each time. The values shown are averaged over 3 times. We see a similar pattern of changes of transfer rate to the fig. 2. There seems no significant difference in transfer speed between writing to one CFS file and writing to several CFS files.

4. Block Write to Distinct CFS Files

The write we did above is to put all the bytes we want to write to a CFS file in one cwrite (concurrent write) statement of a node program. It is interesting to see if there is any change in transfer speed if we break the whole size of writing into samller sizes. In this test, we let each node write a total of 6 megabytes to its own file. We add an addtional DO loop to make each loop write a certain number of bytes (which we call a block). The curves seem to show no significant difference compared to previous non-blocked writes. Here for each node, the T_i is the sum of the the times it took to complete its block writing. When we used a further smaller block size, e.g. 4k bytes, the transfer rate was seen decreased. We suspect this may be caused by the overhead in the start or end of the cwrite function.

5. Writes To CFS Files With I/O Node Allocation

We know the system has 10 I/O nodes. we have so far assumed we used all available I/O nodes on the system. Next, we would like to see what happens if we restrict the number of I/O nodes to use in our test. We assume that the allocation of I/O nodes can be done by using a system call 'restrictvol'. The function 'restrictvol' takes an integer array argument whose components specify the index of volumnes (which we assumed have one-to-one correspondence with I/O nodes) one want to use. In figure 5, we tested write to one CFS file on the whole cube of 128 nodes. We show comparisons between results of using restrictvol call and the result without using restrictvol call. The total bytes each node writes is 6 megabytes. We made tests of restrict volums to 1 and 5. The values are an average of 3 times. As we can see from the figure, almost no change occured. Figure 6 is a similar test of write to distinct CFS files. We have no explanation of why the restrictvol call has no effect on the CFS I/O speed.

Fig. 1: Single Node Write to a CFS File

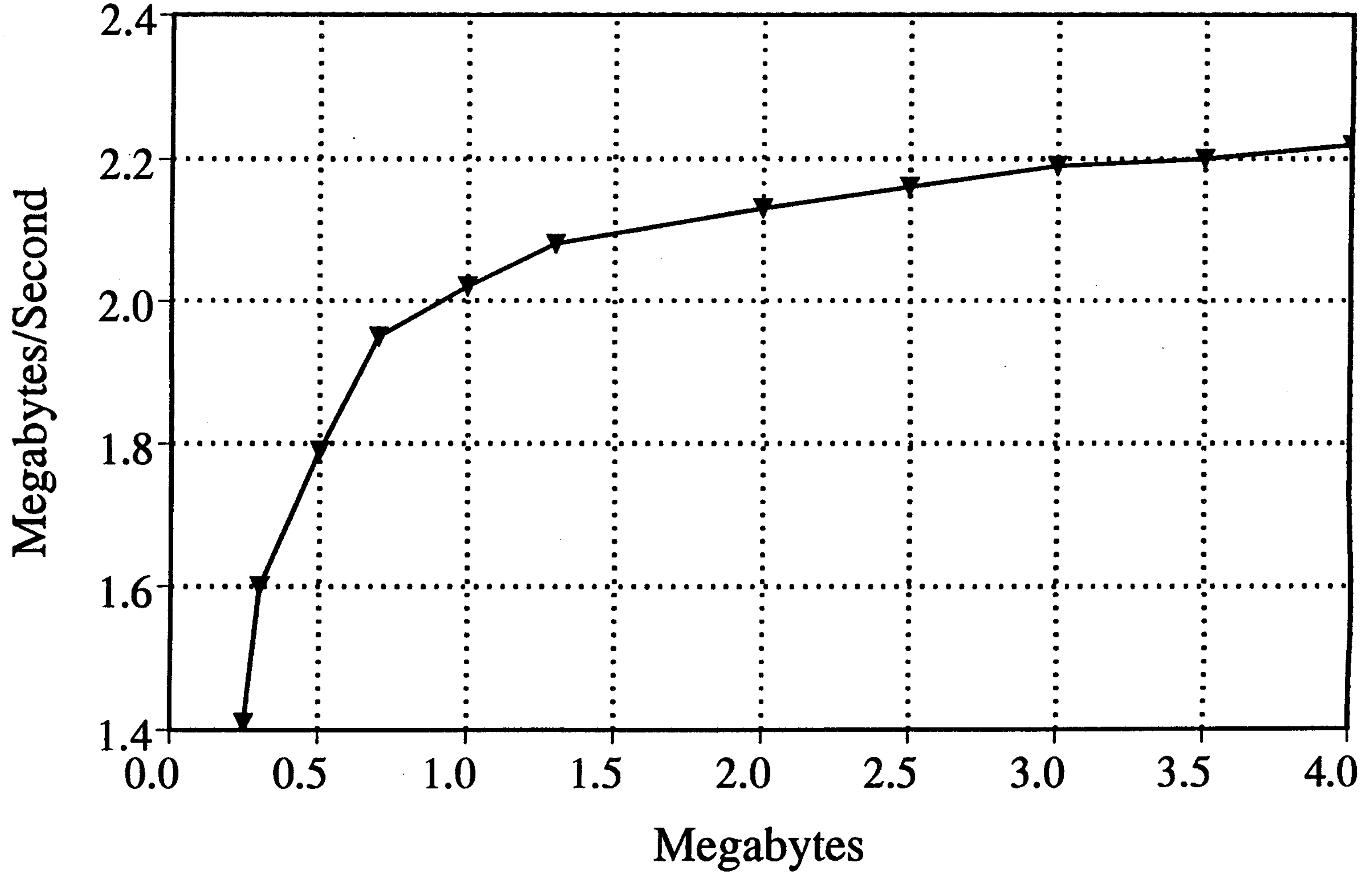


Fig. 2: Multiple Node Write to One CFS File

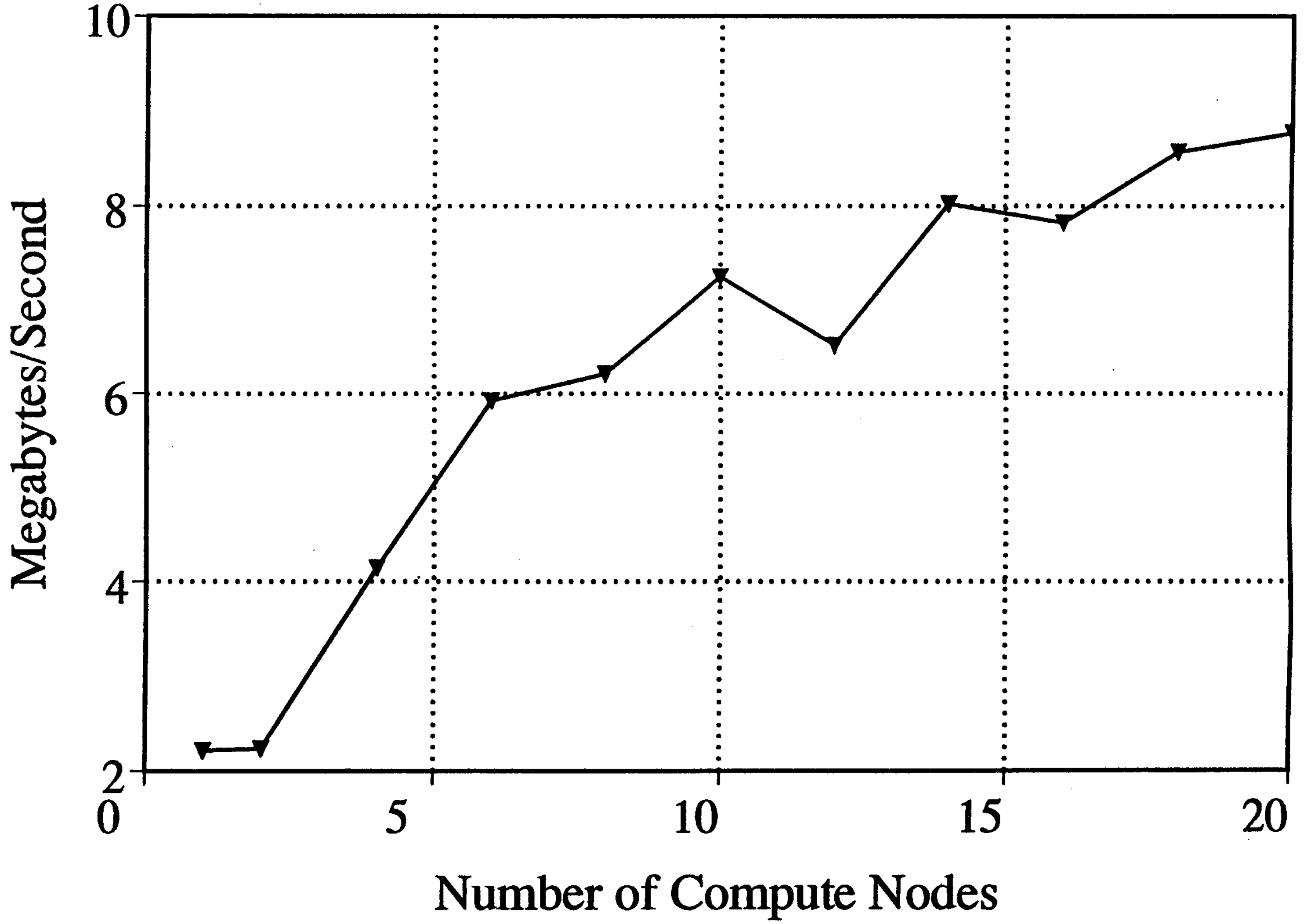


Fig. 3

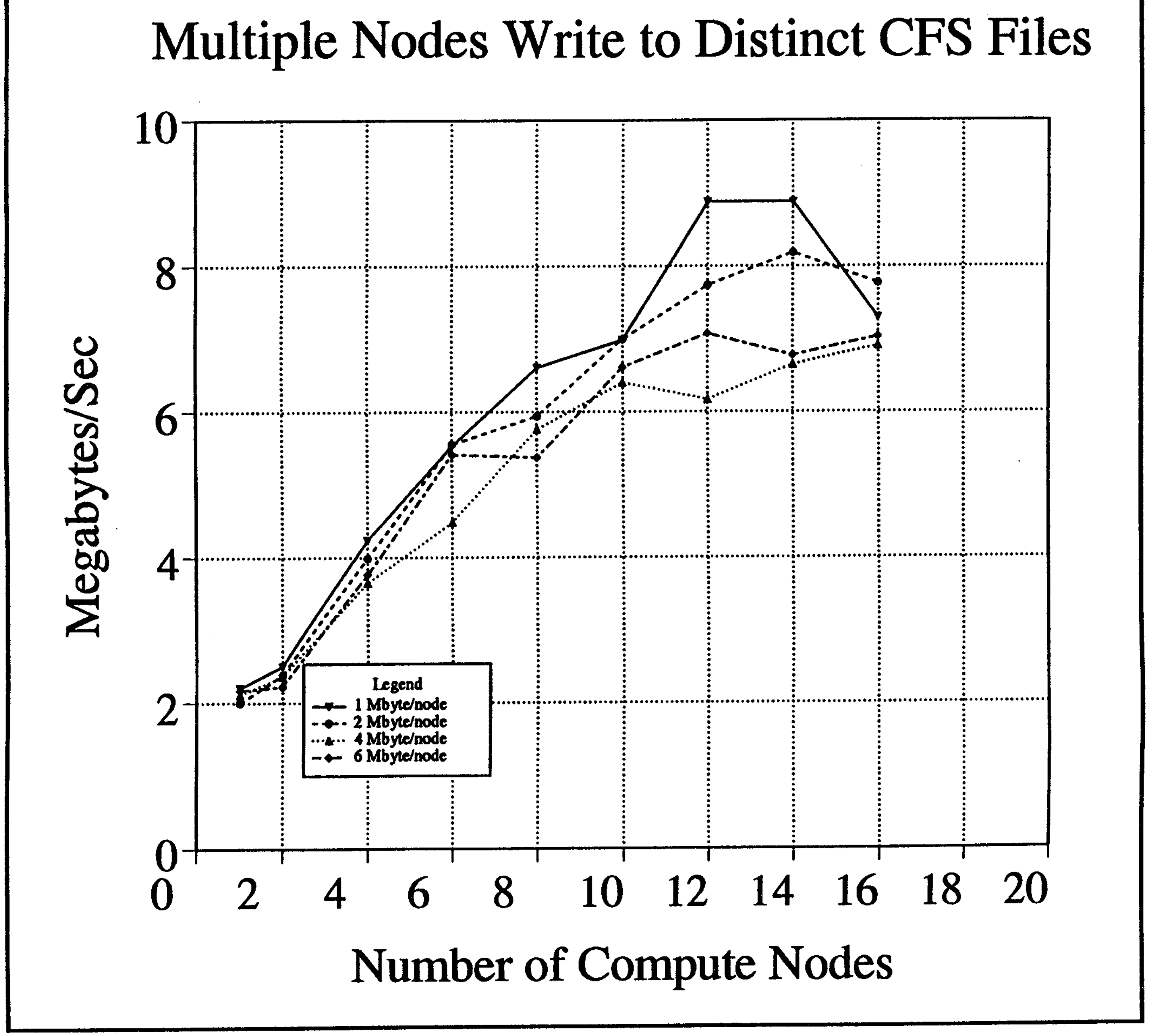

Multiple Nodes Write to Distinct CFS Files
Megabytes/Sec
0
2
4
6
8
10
Number of Compute Nodes
0
2
4
6
8
10
12
14
16
18
20
Legend
1 Mbyte/node
2 Mbyte/node
4 Mbyte/node
6 Mbyte/node

Fig 4: Block Write to Distinct CFS Files

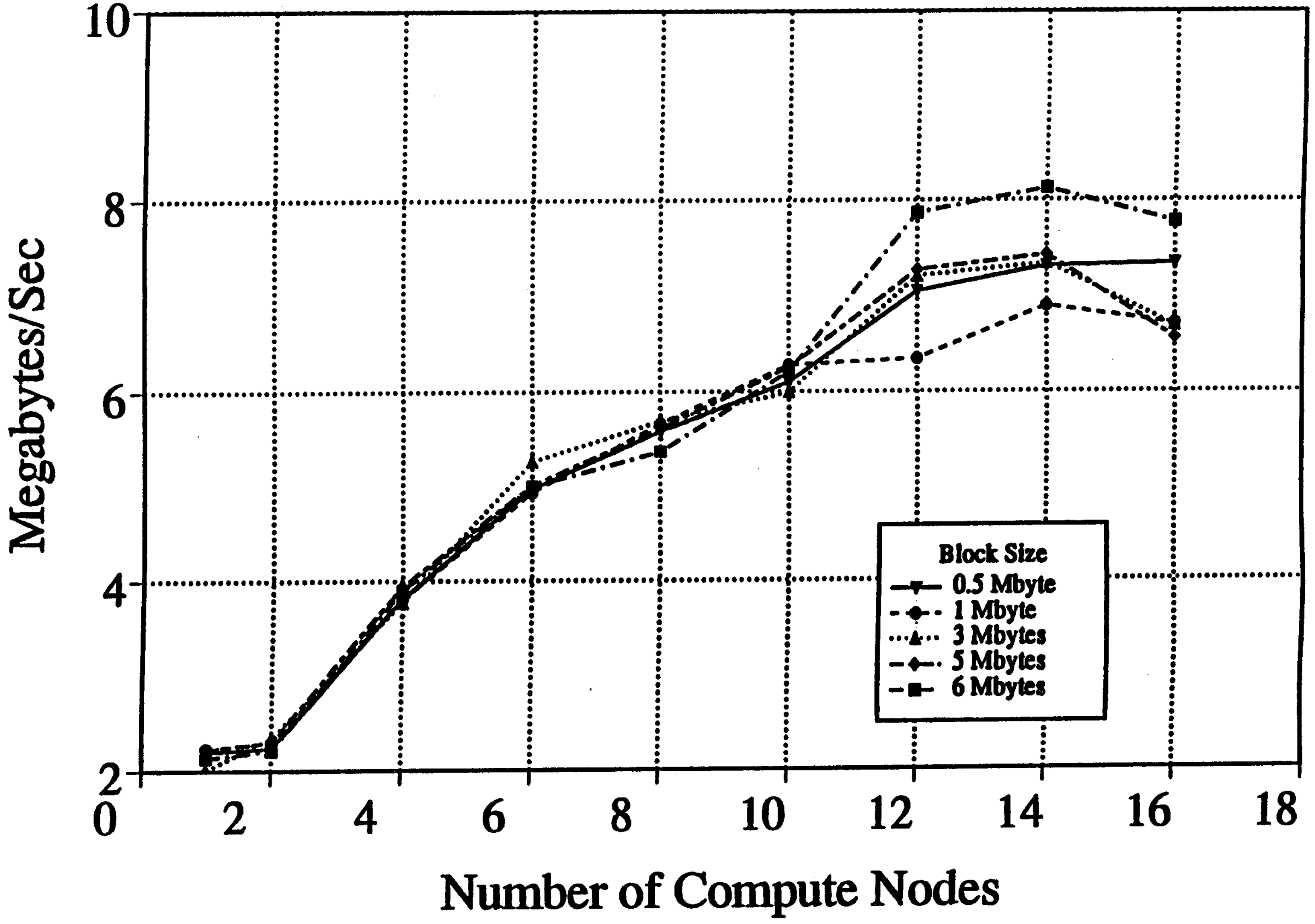
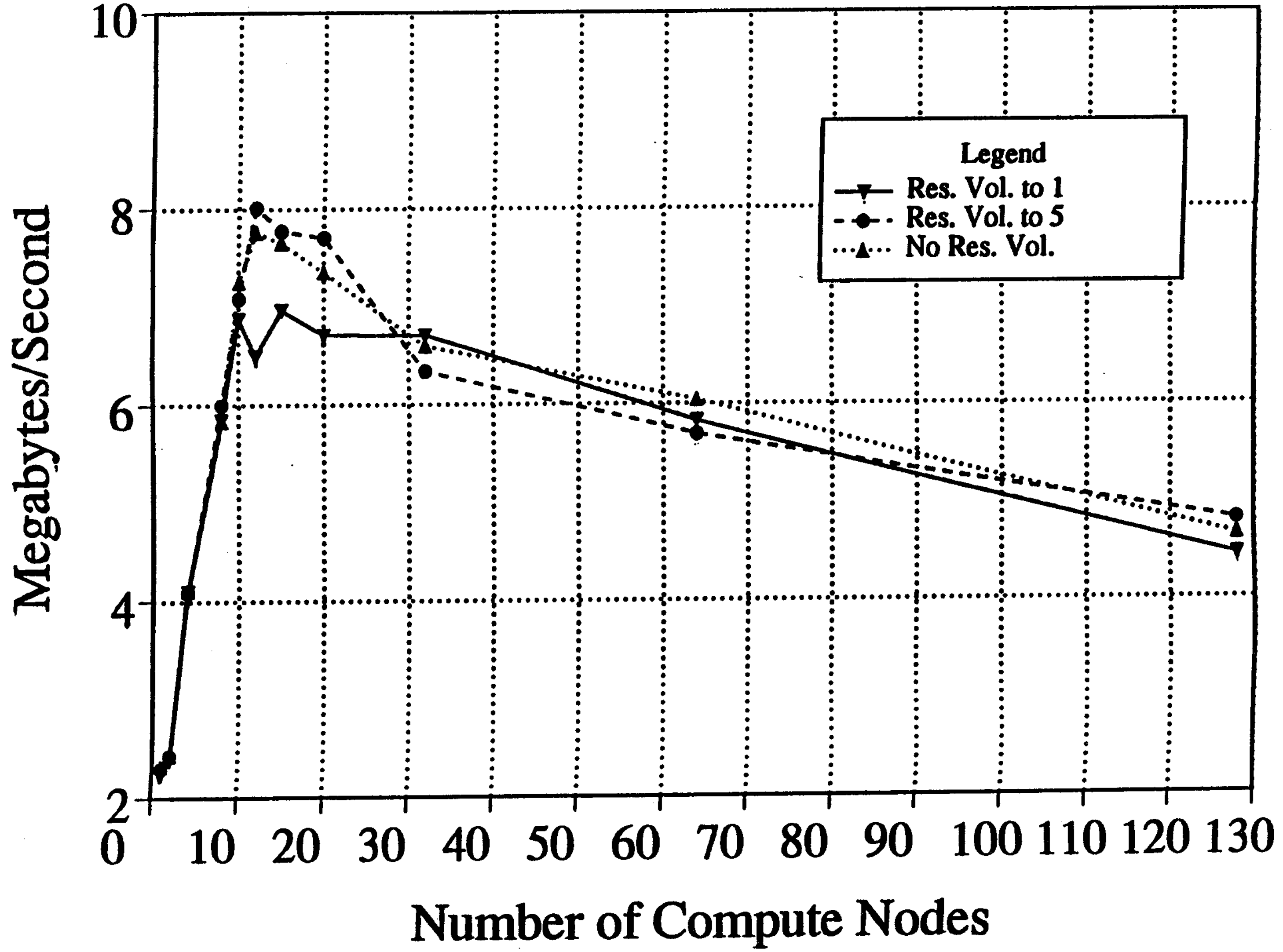


Fig. 5: Multiple Nodes Write to One File


Legend
Res. Vol. to 1
Res. Vol. to 5
No Res. Vol.
Megabytes/Second
Number of Compute Nodes